Caution.

Currently not all the LEDs extinguish as the clock cycles through modes. The cause must be something in the code. Graham thinks it is a memory issue in the Arduino. The test code for the Charlieplex works without issue - see later in these instructions. So if you built it please bear this in mind. This caution will be removed once the bug has been fixed. If you want to solve it first then please report back.

Instructions for building and setting up the Chlonos clock. A project that is described at https://code.google.com/p/chlonos/

These instruction relate to the construction of the Chlonos clock that is listed on the Google Code page at https://code.google.com/p/chlonos/

The clock was conceived, designed and constructed by Graham Richter.

The purpose of these instructions is to provide a step by step guide to enable you to build the clock.

For details as to what the clock does and how it works please see the page at https://code.google.com/p/chlonos/ Also note the video.

Skill set.

There are some basic skills that are required.

Ability to solder competently and to recognise dry joints.

Access to a computer with software to run the Arduino environment and to upload and compile software in the Arduino environment. This is explained later in these instructions.

Some basic understanding of ‘commenting out’ in program source code. This is also explained later in these instructions. Higher knowledge of programming is not required.

Ability to identify components and their values.

Basic understanding of Arduino and prototype shields. A good book that can be recommended is Arduino for Dummies.

Work process.

A clear work space where solder fumes are not likely to irritate you or others who share the accommodation.

Freedom from distraction.

Component list.

1 x Arduino Uno - Try here
1 x LoL Shield - These are in two types. The one you will mostly likely find is the one that is the same size as the Arduino. Fine if you want a small clock but the project uses the Olimex Shield LOL. LOL in this context stands for Load of LEDs which can be seen on the website www.olimex.com. This board measures 110 x 166 mm. It consists of an array of 118 10mm LEDs. in an array 9 x 14. They come in white, blue, green or red. You may still be able to find a kit of

parts and build your own but it does involve a considerable amount of soldering. The LOL Shield was obtained direct from Olimex.

1 x Speakjet Speech Synthesiser chip This is a 18 pin chip.
1 x LM386 Audio Amplifier chip. These were sourced as a pack of 10 for £5.92 from Amazon, UK supplied by leading-star
1 x 8ohm 0.3W speaker
1 x N55AY light dependent resistor
1 x DS1307 Realtime Clock chip - You can either buy the individual components such as the pull-up resistors, crystal, chip, battery backup holder, etc, or you can get it all for £2 assembled on a PCB from Ebay
1 x IN914 signal diode - this will go between the Speakjet chip's 'busy speaking' pin and the audio amp pin 7 to switch it off when you're not using the speech. Otherwise the LoL Shield will generate a ton of noise.
1 x Arduino Prototype shield - acts as a project board that plugs directly into the Arduino. Very handy. Ebay has several.
1 x LM35 celsius temperature sensor.

Five resistors with the following values
R1 27K
R2 27K
R3 120
R4 10K
R5 10

Seven capacitors
C1 10µF Tantalum.
C2 470µF electrolytic
C3 0.01µF Ceramic you will find body marked 103 the last digit indicating the numbers of zeroes.
C4 1µF Tantalum
C5 0.047µF Ceramic you will find body marked 473 the last digit indicating the numbers of zeroes.
C6 0.01µF Ceramic you will find body marked 103 the last digit indicating the numbers of zeroes.
C7 2.2µF electrolytic

A push to make switch.

Two sockets. One for the 18 pin Speakjet chip and one for the the 8 pin amplifier. You can of course directly solder the chips to the board but if you need to replace one for any reason it is difficult and so socketing is the way to go. If you cannot find an 18 pin socket a longer one will do and you can just cut it down. A Dremel is excellent tool for this.

Some single strand signal wire
1 x 9volt 2.1mm power supply. Centre pin positive
1 x white enclosure - For some reason they produce the LoL Shield with clear directional LEDs, not diffuse ones. An enclosure with semi-opaque white covering will disperse the light nicely. Alternatively, a blank sheet of paper over the LEDs will work too.

Many parts can be sourced from Maplin. www.maplin.co.uk

It is well worth shopping around.

Instructions.

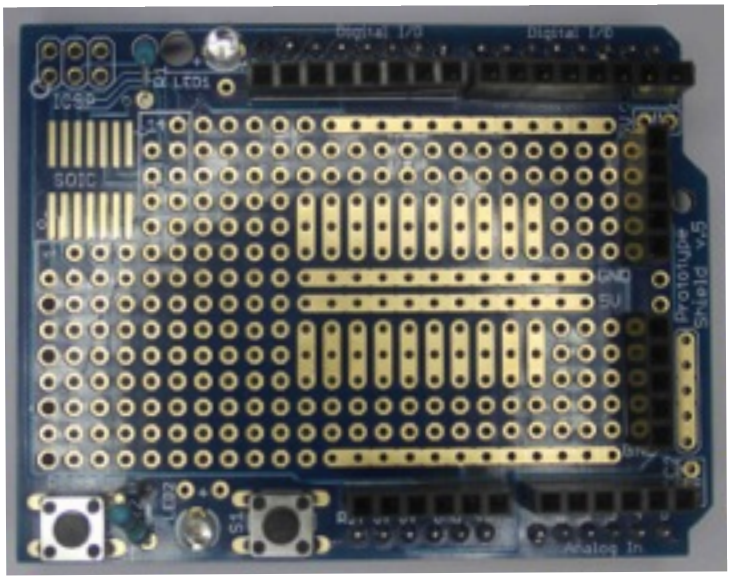
The construction consists of three main components that piggyback together. At the bottom is the Arduino next is the Prototype shield board and last is the LOL Shield. These three items physically link together. The LOL Shield is connected by the digital pins 0 to 13 and Gnd and Aref.

Construction concentrates on the Prototype shield. The connections and peripheral components are connected to the Prototype shield.

There are two integrated circuits on the Prototype shield. The first is the Speakjet chip which is an 18 pin chip with nine pins on each side. There is a cutout at one end. By convention the pins are number anticlockwise if you hold the chips with legs pointing way form you.

The other chip is the LM386 this is also a dual inline chip but has 8 pins four on each side.
You could solder the chips directly to the Prototype shield but this will give you trouble if there is a fault and you need to replace the chip. So installing a socket is the best way forward.

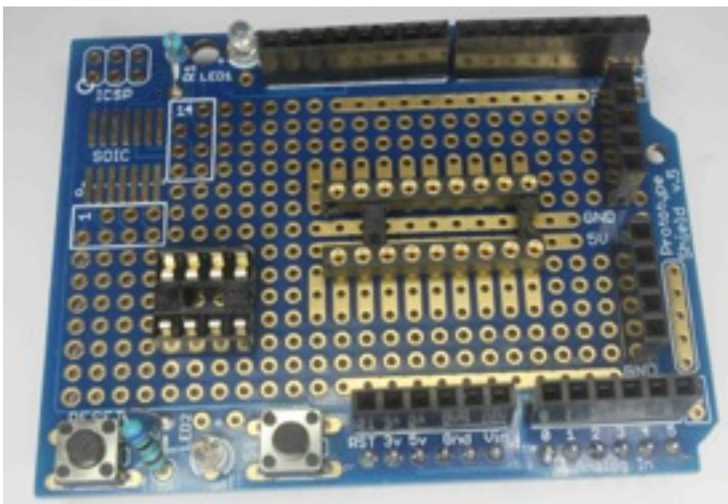
Here is a picture of the Prototype shield as you receive from the supplier.



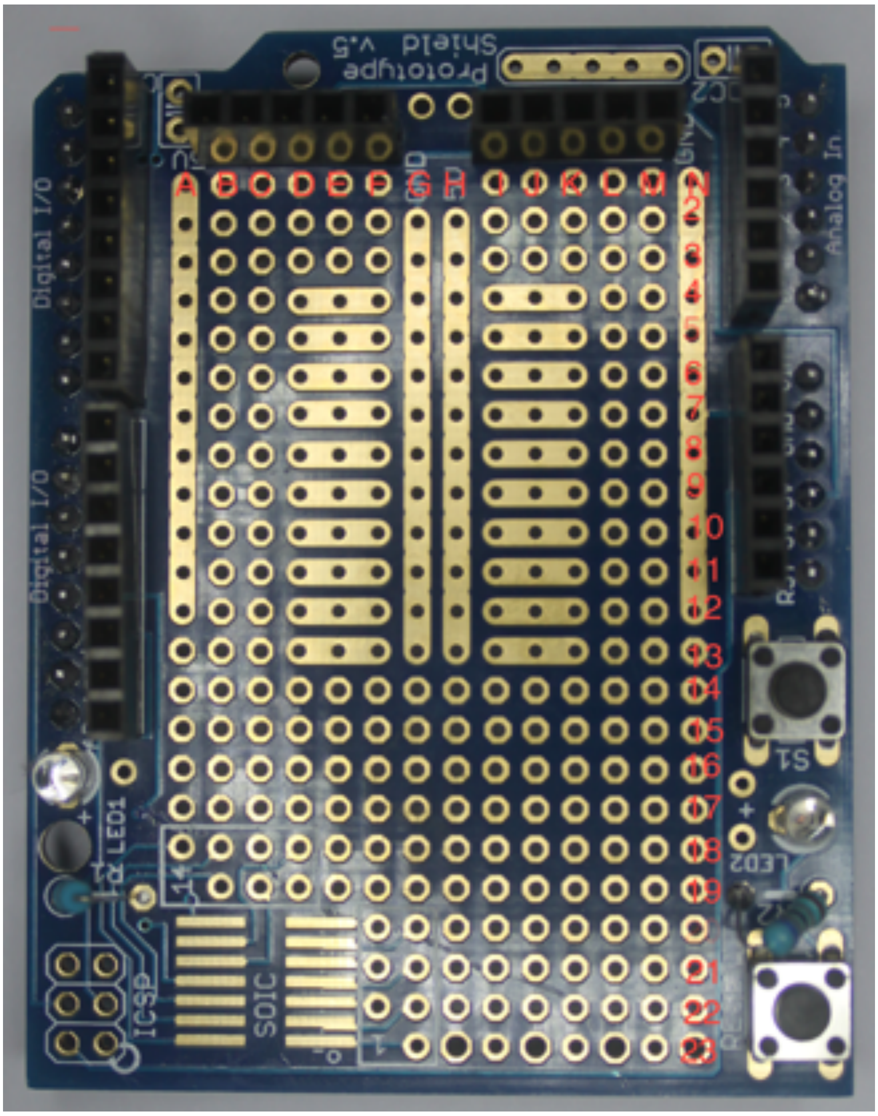
Begin first by placing the two sockets in the position shown in the photo. Note each socket has a half circle cutout at each end. It is to the right hand side in these photos. It is used later to orientate the chips. Turn the board over and solder pin 1 and pin 9 of the socket for Speakjet chip and pin 1 and pin 5 of the chip. The purpose of this is to hold the sockets in place while other components are added.

You will probably find the sockets drop out as your turn the board over. Use a small piece of Bluetack at the ends of each socket. This will enable you to solder the socket in place. Remove the Bluetack.

The next stage is to add a number of links to the board.



I had a problem with describing this as I have not seen in any of the Arduino Documentation any way of describing the individual contact points on the board. Of course the points to not contact and are just plated through holes unlike purpose designed printed circuit boards. However, without a reference system it does pose a problem if you need to describe a board, as I do here. I have a devised a simple system of lettering the columns and numbering the rows which you will see in the next photo.



The columns are lettered left to right from A to N. There is no point in letter the sockets as they have unique references. The rows are numbered from 1 to 23. By coincidence Column G is Ground at least from G2 to G13. So adopting this reference system makes it an easier task to describe where links and components should go.

Links.

I have used wire with a white covering so the links stand out.

Install the following links

Analog socket Pin 2 to B5.

Analog socket Pin 1 to C19

Analogue socket Pin 0 to J12

Hole J8 to hole J10

Hole K6 to hole M14

Hole K9 to hole K10.

Hole M2 to hole H17

Hole N9 to hole K9

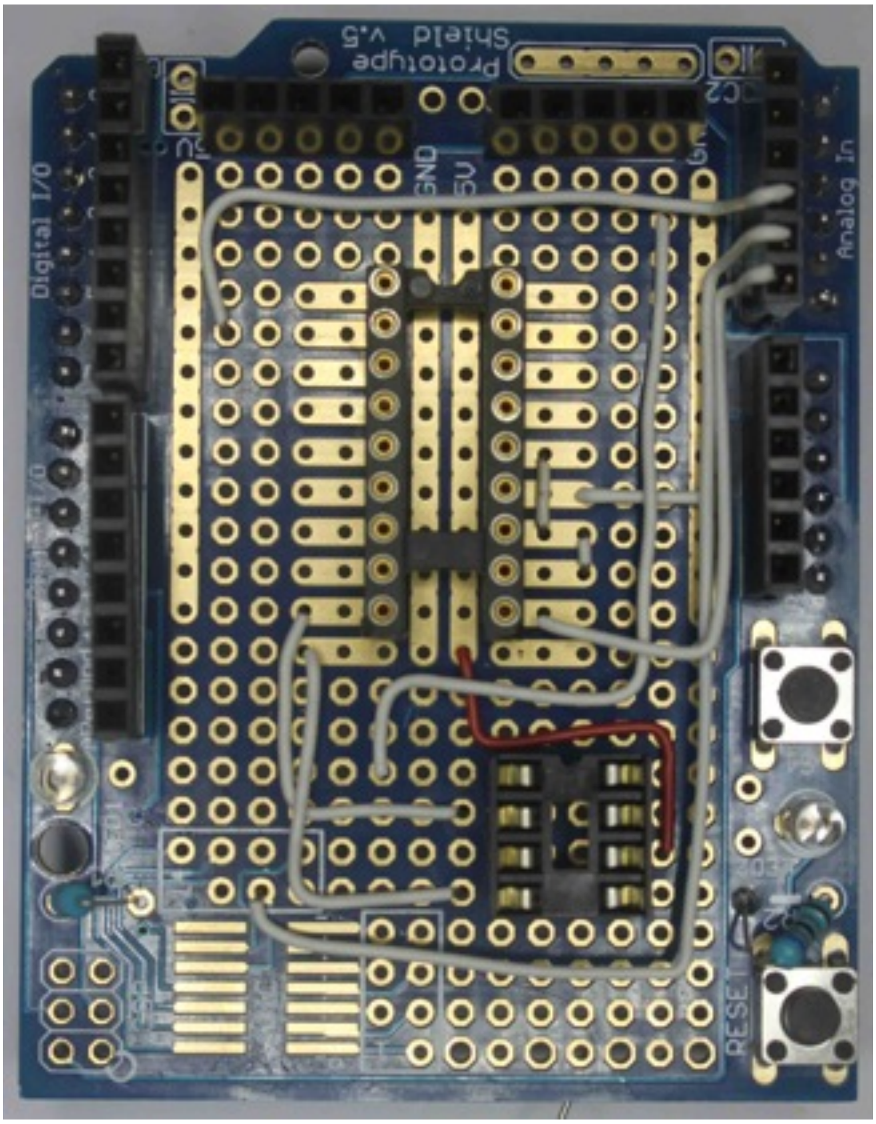
Hole H13 to hole M18 this is shown as a red wire in the next photo.

Hole D12 to hole H17

Hole D13 to hole H19

It should look like this. See photo.

The next task is to insert the passive components.



Resistors

R1 is inserted in holes K4 and L4

R2 is inserted in holes L1 and L3

R3 is inserted in holes C6 and D6

R4 is inserted in holes F20 and G20

R5 is inserted in holes F21 and I21

Capacitors

C1 is a tantalum capacitor which is polarity conscious the negative lead is identified by two '- -' that lead goes to N19. The other lead to N22

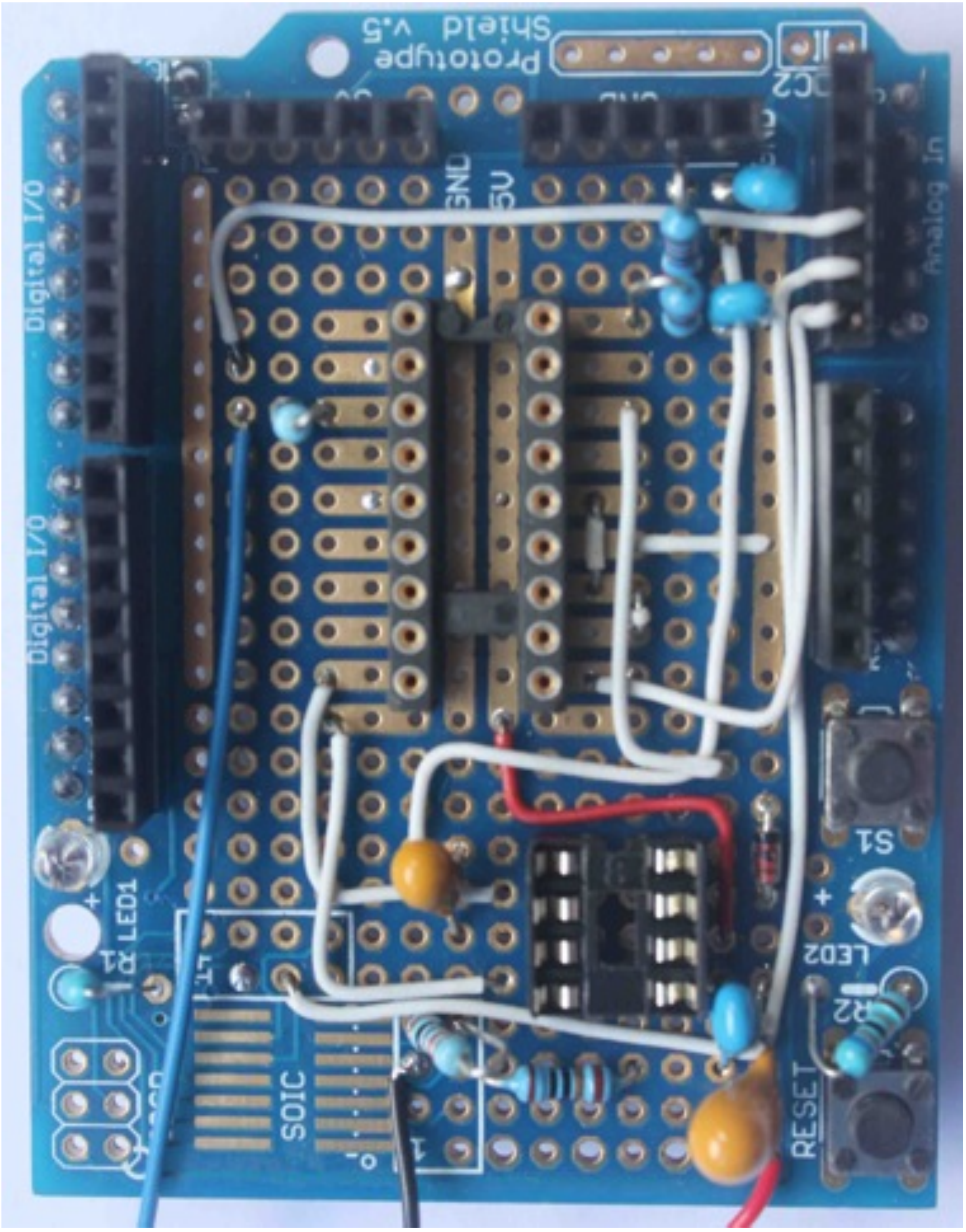
C6 is inserted in holes M4 and N4

C3 is inserted in holes M1 and N1

C4 is a tantalum capacitor which is polarity conscious the negative lead is identified by two '- -' that lead goes to G16. The other lead to G18.

C5 is inserted in holes M19 and M21

The prototype board now look like this.

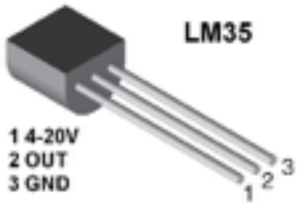


Peripherals.

There are a number of external links. These comprise the loudspeaker, the RTC clock module, the LM35 temperature sensor, the push to make switch and the light dependant resistor.

These are connected as follows:-

Loudspeaker. If the loudspeaker has a + sign then attach a lead to that and place the end of the wire in hole N23. Connect a lead to the other tab on the loudspeaker and that is connected to the ground strip in hole N12.



The LM35 looks like this.   Identify the Ground and Positive voltage pins. The centre pin is the output. Connect the positive voltage pin to hole A11. In the photo of the completed board at the bottom on the page it is coloured red. The output pin coloured blue is connected to Analog socket 3. The Ground wire coloured black is connected to hole N13.

The push to make switch is connected to holes A12 by a red lead and to hole F11 by a black lead.

The light dependant resistor is connected to hole A6 by a red lead and to B6 by a blue lead.

The last component is the RTC. There are four connections to the Prototype shield. Ground connects to hole G2. The positive supply is connected with red wire to H2. SDA connects to Analog socket 4 with a blue wire. SCLI connects to Analog socket 5.

There are two more capacitors to connect. C2 connects to the top right of the board and overhangs. The Prototype shield is likely to be marked in white as C2.   Not that this a polarity conscious component and the the capacitor is identified with a negative sign next the lead that is negative. That goes into hole in line with the analog socket. The lead is next to it.

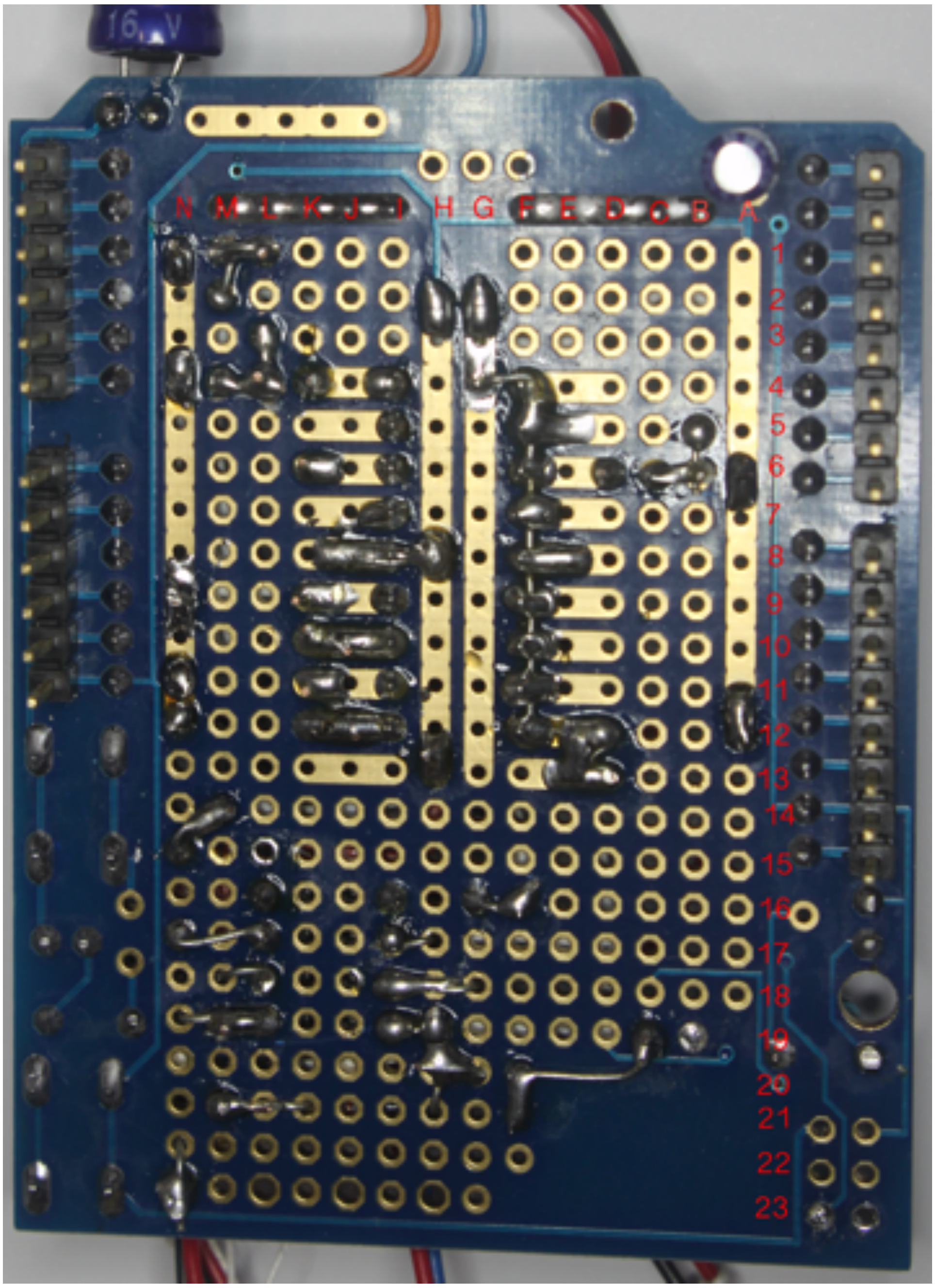
The last capacitor C7 is fitted on the under side of the board and as can be seen in the photo where the board is is shown solder side. This is also polarity conscious and the negative lead goes nearest to the edge of the board.

Soldering

With all the components now in place it is time to solder them. You may need to take care as when turning the board over is it easy for links and components to drop out. One method you can try is the use a soft sponge pressed against the components side of the board so that when it is flipped over everything stays in place.

Links to components are made by using the surplus leads and bending them over the the places where they connect.

The same lettering and numbering system is used as was used to describe where the links and components were placed. Note the lettering is now reading from right to left and not left to right. This is to avoid confusion. Numbering is unchanged. Referring to the next photograph proceed as follows:



Solder N1
Solder together M1, L1 and M2
Solder H2
Solder G2
Solder N4
Solder together L3, M4 and L4
Solder I4
A link is made joining pins F4 through to F13. I missed the connection between F12 and F13 but a piece of wire needs to laid in column F to connect pins F4 through to F13. Note the wire terminates in G4.
Solder I5
Solder B5
Solder K6
Solder I6
Solder D6
Solder together C6 and B6
Solder A6
Solder I7
Solder J8, I8 and H8 together
SolderF8 and E8 together
Solder N9
Solder K9
Solder I9
Solder the connection in pad K10, J10 and I10
Solder D13

Solder D14
Solder together N15 to M15
Solder L16
Solder I16
Solder G16 together G16 and F16
Solder together N17 to L17
Solder together I17 to H17
Solder M18 to L18

Solder I18, H18 and G18 together
Solder N19, M19 and L19 together
Solder I19, H19, H20H21 and G20 together
Solder F20, F21 and C19 together
Solder M21 to K21
Solder N22 to N23

Once you have done this it should look like the photo save for the link I missed between F12 and F13. See above.

Solder the capacitor C2 and the one of the one that can be seen near A C7.

The soldering is done.

Check for dry joints and anything that you have missed.

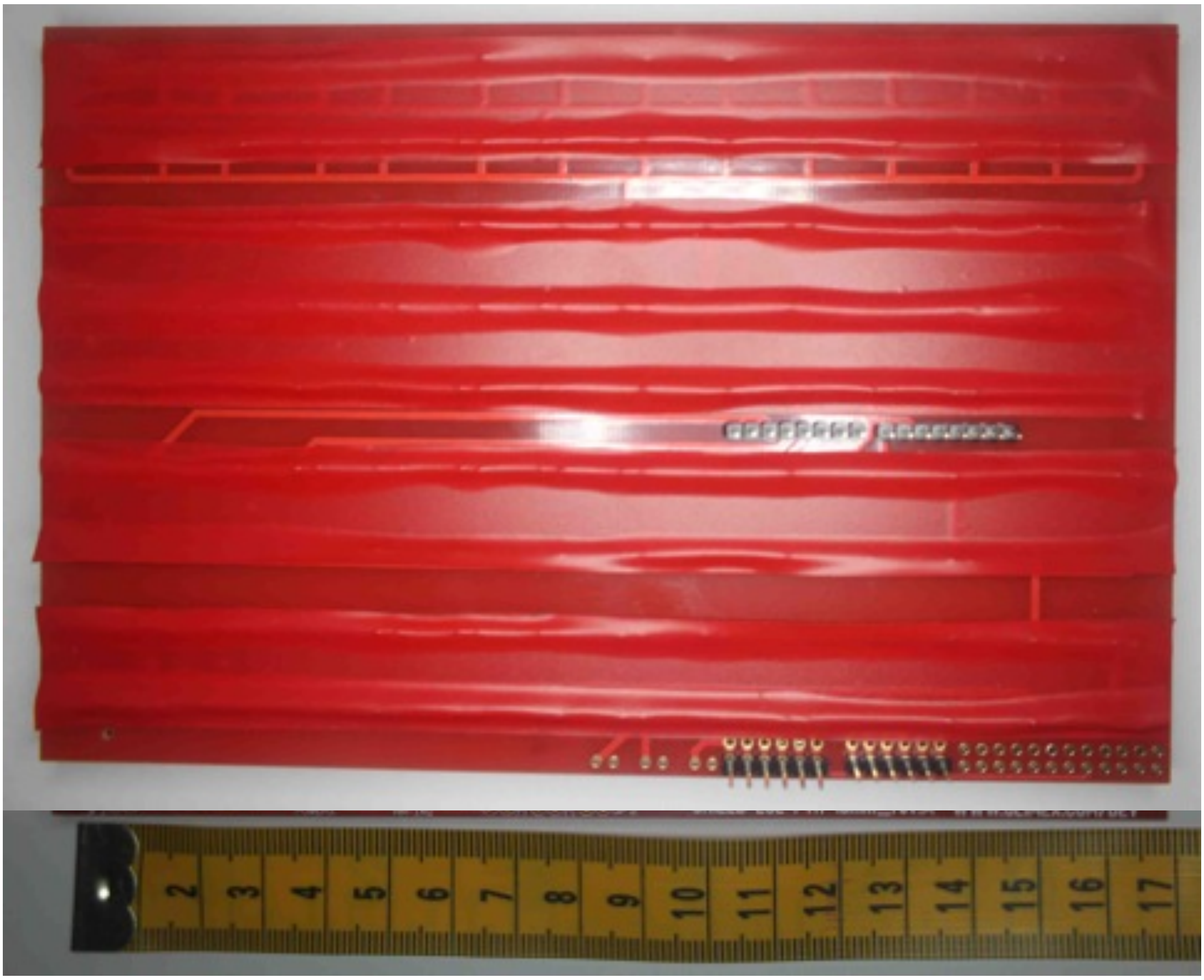
Chips.

The two chips can now be inserted. I usually find that the pins are a little too wide to fit into the sockets. Lay the chip down with the pins laying against a flat surface and gently bend first one set towards the centre of the chip. Turn the chip round and do the other side. You should now find that the each chip will fit into its respective socket. Not the cut out on the sockets must match the cut on the chips.

You are now ready to assemble the three components

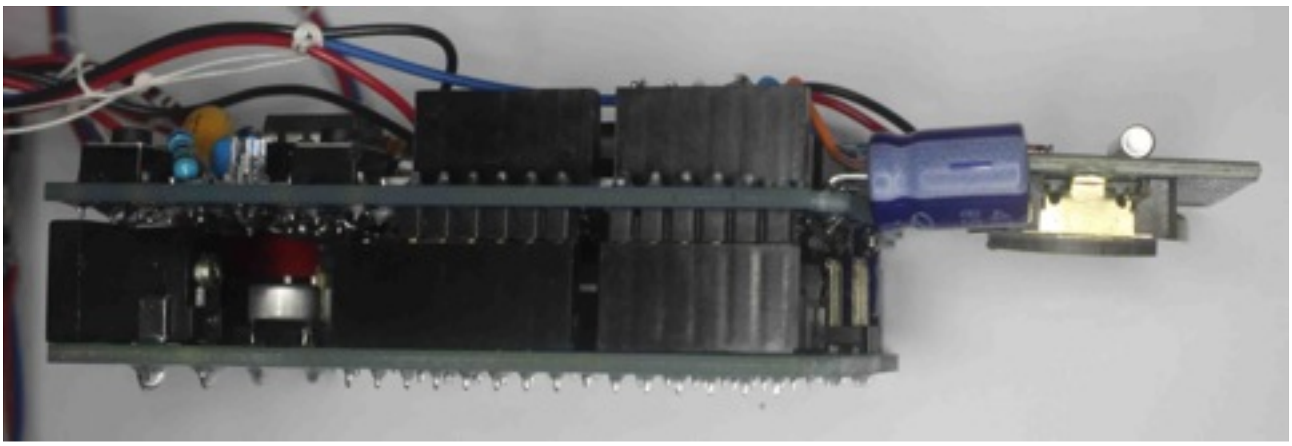
Assembly.

The LOL Shield comes with two sets of pins. The set of pins towards one edge do not do anything and are not connected. It is the 16 pins in the centre that are connected to the LEDs.



The pins on the back of the LOL shield could easily short if they touch other components. So add some insulation tape and tape them off.

Similarly the USB socket on the Arduino comes very close to the Prototype shield and that too should have a piece of tape on top.

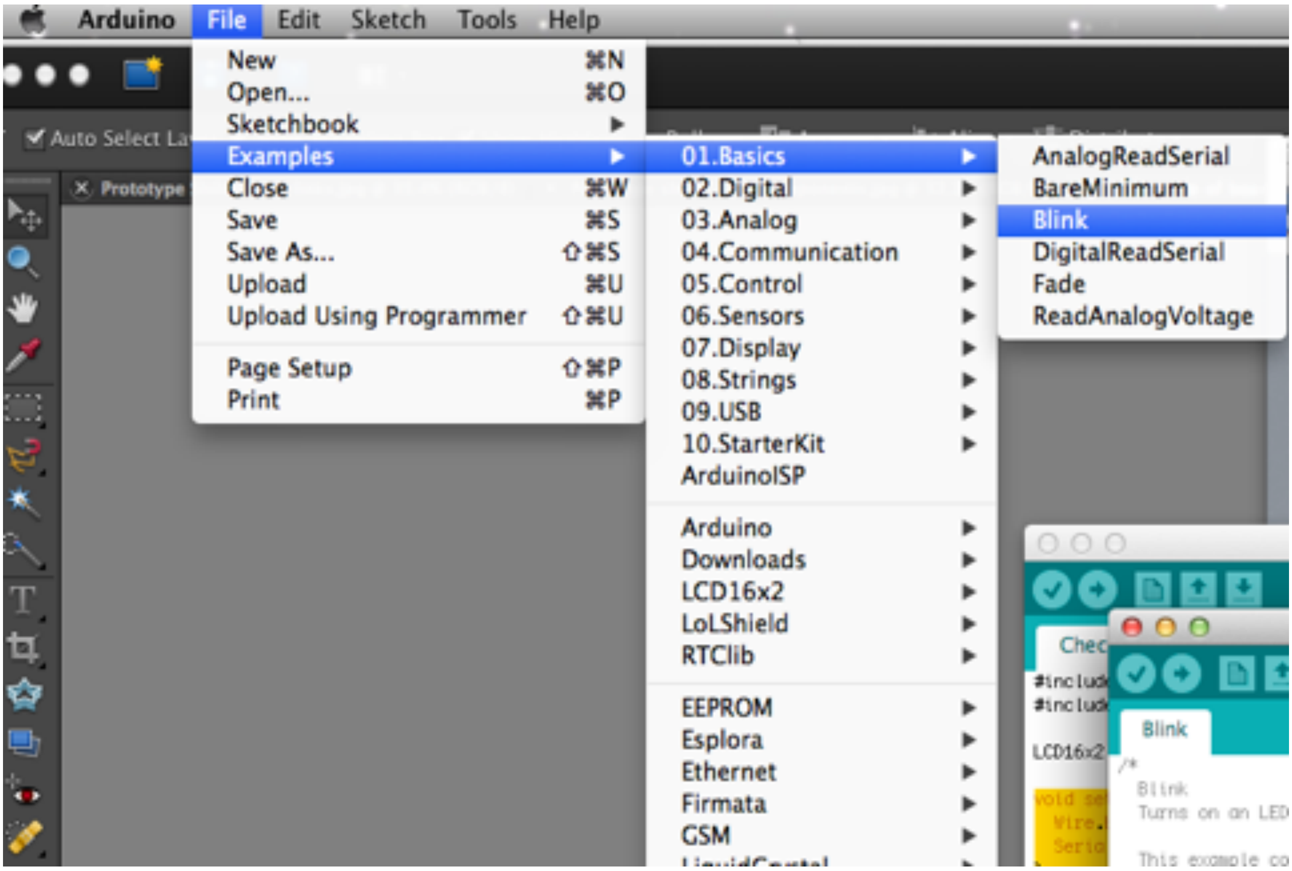


First connect the Arduino and Prototype shield together. It is difficult to get these boards the wrong way round. Note one end of the Arduino has rounded corners. These match the Prototype shield. See photo.

Now add the the LOL Shield. This can connected the wrong way round. See the photo for orientation.

Nearly there.

If you have not already done so install the Arduino software on your computer. There are versions for PC, Mac and Linux. Connect the USB socket on the Arduino to you computer USB port. It is worth checking that the Arduino is working. Some Arduinos come with a small program that is called Blink. Once the Arduino is connected to the computer the LED on the Arduino near Pin 13 should flash once a second. If it does not then go to File, Examples, 01.Basics, Blink and selected it.



Once selected you will see this page

This is the Blink program. If you have a familiarity with programming then it is akin to the “Hello World” programs that you come across.

Click the the “Tick” in the top left hand corner and the code should compile without any errors. Next Click the right facing arrow next to the “Tick”. This will upload the code to the Arduino and if all is well the little LED should flash once per second. If this does not happen if could be that you board is missing the LED. If so add a LED between Pin 13 and ground and see if it flashes. Note that it will only work if the cathode goes to ground.

If all is well then it is time to turn attention to the code. The code can be found here.

https://code.google.com/p/chlonos/downloads/list

Open it in the Arduino software and you will see the code.

Note these lines of code.

#include <Charliplexing.h> // Library for driving the LEDs
#include <Wire.h>
#include <RTClib.h>
#include <SoftwareSerial.h>
#include <avr/pgmspace.h>
#include <EEPROM.h>

These are libraries that the Code uses. Some are built in but you will need Charlieplexing and RTClib.

Charlieplexing can be found here.

http://playground.arduino.cc/Code/Charlieplex

You will also need the RTC library which you can find here.

https://github.com/adafruit/RTClib

One caution. When you download the RTClib you get a file called "RTClib-master". Arduino will complain if you try to install it. So rename the file to "RTClib.zip" and it should now install without error.

Install both of these. If they have installed correctly then when you see them listed on the tab Sketch, Import library as 'contributed.' The other libraries are already built into the Arduino platform.

Now compile the code for the project that you downloaded.

It may come up with an error. If so find this line in the code.

RTC.adjust(DateTime(__DATE__, __TIME__),16)

Change the line to this

RTC.adjust(DateTime(__DATE__, __TIME__))

and recompile. It should now compile without error.

Now Upload the code and when it finishes, about 10 seconds, the clock should burst into life.

Congratulate yourself.

It is very unlikely the time is correct so you need to set the time. Proceed to the section of Setting the Time.

Setting the Time.
You may have wondered on reading up about this project how the the time is set. There is only one button but that does not set the time. The time is stored in the RTC module which runs when the power it off. It has it own button cell battery.

So to set the the time you need to go back to the code.

Find the line

RTC.adjust(DateTime(__DATE__, __TIME__))

It is this line that reads the time and date from your computer clock and inputs it to the RTC.

```
// Set the RTC time to match compiler time. Comment out after the clock has been set once and recompile.
  //RTC.adjust(DateTime(__DATE__, __TIME__)); // Set RTC date to match compile time and turn on 1HZ output for LED (bit 4 on the control register)
  if (RTC.isrunning()) { // check if Real Time Clock is running
    //loadSentence(31);// Say "setup-ok"
    //SerialSpeakjet.print(sentencebuffer);
  } else {
    //RTC.adjust(DateTime(__DATE__, __TIME__)); // Set RTC date to match compile time and turn on 1HZ output for LED (bit 4 on the control register)
  }
```

You twill see that the lines with the // are greyed out. They are only needed to set the time. The two forward slashes are used for comments so that the when the Arduino encounters them it ignores them. Without the // the line in considered to be an instruction. So delete the forward slashes shown in red and the lines will then be interpreted as instructions. They cease to be greyed out.

Compile the code and upload to the Arduino. You should hear the voice say "Set up ok." The time is now stored in the RTC.

Now go back to the code and put back the forward slashes. Compile the code again and upload it.

The time is now read from the RTC.

Job done. You have a working clock and can leave it as it is or play with the code if you wish.

The task of finding or making a suitable enclosure is up to you.

Trouble shooting.

If you built the Display yourself or suspect that not all the LEDs are working then you can test it.

Go to

https://codebender.cc/example/Charliplexing/Basic_Test

Download the code. It causes a pattern to be played across the LOL Shield. LEDs that are not working will be obvious.

Finally.
If there are problems then post a question to the Chlonos page.

Simon S. Allen E & OE 2nd. May 2014. Version 1.0